

[Home](#) | [Login](#) | [Logout](#)


Welcome United States Patent and
Trademark Office

Search Results

[BROWSE SEARCH](#) [IEEE GOLD](#)

Results for "(((slicing and optimiz* memory management)<in>metada1
(pyr >= 1995 <and&w
Your search matched 0 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevance
Descending order.

» Search Options

[View Session](#)
[History](#)
[New Search](#)

Modify Search

☐ Check to search only within this results set

» Key

**IEEE
JNL**

IEEE

 Journal or
Magazine

**IEEE
JNL**

 IEEE Journal
or Magazine

**IEEE
CNF**

IEEE

 Conference
Proceeding

**IEEE
CNF**

IEEE

 Conference
Proceeding

**IEEE
STD**

IEEE

Standard

Display ☒ Citation ☐ Citation &
Format: ☐ Abstract

No results were found.

Please edit your search criteria and try again. Refer
assistance revising your search.

Indexed by



[Home](#) | [Login](#) | [Logout](#)
IEEE Xplore[®]
RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results

[BROWSE SEARCH](#) [IEEE GUIDE](#)

Results for "(((slicing unused objects)<in>metadata)) <and> (pyr >= 19 pyr <= 200...

Your search matched **0** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance Descending** order.

» Search Options

[View Session History](#)
[New Search](#)

Modify Search

(((slicing unused objects)<in>metadata)) <and> (pyr >= 19 pyr <= 200...

☐ Check to search only within this results set

» Key

IEEE JNL IEEE
Journal or
Magazine

IEEE JNL IEEE Journal
or Magazine

IEEE CNF IEEE
Conference
Proceeding

IEEE CNF IEEE
Conference
Proceeding

IEEE STD IEEE
Standard

Display Format: ☒ Citation ☐ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by
Inspection

[Home](#) | [Login](#) | [Logout](#)


Welcome United States Patent and Trademark Office

Search Results

[BROWSE SEARCH](#) [IEEE GUID](#)

Results for, "(((slicing class memory)<in>metadata)) <and> (pyr >= 199 <= 2002..."

Your search matched **0** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance Descending** order.

» Search Options

[View Session](#)
[History](#)
[New Search](#)

Modify Search

(((slicing class memory)<in>metadata)) <and> (pyr >= 199 <= 2002...)

☐ Check to search only within this results set

» Key

IEEE JNL

IEEE

Journal or Magazine

IEEE JNL

IEEE Journal or Magazine

IEEE CNF

IEEE

Conference Proceeding

IEEE CNF

IEEE

Conference Proceeding

IEEE STD

IEEE

Standard

Display Format: ☒ Citation ☐ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by



[Home](#) | [Login](#) | [Logout](#)


Welcome United States Patent and Trademark Office

☐ Search Results

[BROWSE SEARCH](#) [IEEE GUIDE](#)

Results for "(((compact*<in>metadata) <and> (slicer<in>metadata) (memory<...</p>
</div>

Your search matched 0 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by Relevance Descending order.

» Search Options

[View Session](#)

[History](#)

[New Search](#)

Modify Search

☐ Check to search only within this results set

» Key

IEEE JNL

IEEE

Journal or Magazine

IEE JNL

IEE Journal or Magazine

IEEE CNF

IEEE

Conference Proceeding

IEE CNF

IEE

Conference Proceeding

IEEE STD

IEEE

Standard

Display Format: ☒ Citation ☐ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by
 Inspec

http://ieeexplore.ieee.org/search/searchresult.jsp?query1=compact*... 6/21/06


[Subscribe \(Full Service\)](#) [Register \(Limited Service\)](#)

Search: ☒ The ACM Digital Library ☐ The ACM Full Text Collection

+slicing +class +hierarchies


[Feedback](#) [Report a problem](#)

Published since January 1995 and Published before September 2002

Terms used slicing class hierarchies

Sort results by

[Save results to a Binder](#)

Try an [Advanced Search](#)

[Search Tips](#)

Try this search

Display results

☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Results

1 [Slicing class hierarchies in C++](#)

Frank Tip, Jong-Deok Choi, John Field, G. Ramalingam

October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM conference on Object-oriented programming, systems, languages and applications OOPSLA '96**, Volume 31 Issue 10

Publisher: ACM Press

Full text available: [pdf\(2.08 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)


This paper describes an algorithm for *slicing* class hierarchies in C++ programs. Given a C++ class hierarchy (a collection of C++ classes and inheritance relations) and a program P that uses the hierarchy, the algorithm eliminates from the hierarchy members, member functions, classes, and inheritance relations that are unused in P , ensuring that the semantics of P is maintained. Class slicing is especially useful when a sliced program P is generated ...

2 [Class hierarchy specialization](#)

Frank Tip, Peter F. Sweeney

October 1997 **ACM SIGPLAN Notices , Proceedings of the 12th ACM conference on Object-oriented programming, systems, languages and applications OOPSLA '97**, Volume 32 Issue 10

Publisher: ACM Press

Full text available:  [pdf\(2.29 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)


Class libraries are generally designed with an emphasis on versatility and Applications that use a library typically exercise only part of the library's result, objects created by the application may contain unused members. An algorithm that *specializes* a class hierarchy with respect to its usage in an application. The algorithm analyzes the member access patterns for P 's variables, and classes for variable ...

3 [Reengineering class hierarchies using concept analysis](#)

 Gregor Snelting, Frank Tip

November 1998 **ACM SIGSOFT Software Engineering Notes**, Proceeding of the **ACM SIGSOFT international symposium on Foundations of software engineering SIGSOFT '98/FSE-6**, Volume 23 Issue 6

Publisher: ACM Press

Full text available:  [pdf\(1.31 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)


The design of a class hierarchy may be imperfect. For example, a class C may have a member m not accessed in any C -instance, an indication that m could be moved into a derived class. Furthermore, different subsets of C 's members may be accessed from different C -instances, indicating that it might be appropriate to split C into two classes. We present a framework for detecting and remediating such design errors. This paper is based on ...

4 [Understanding class hierarchies using concept analysis](#)

 Gregor Snelting, Frank Tip

May 2000 **ACM Transactions on Programming Languages and Systems**, Volume 22 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(433.91 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

A new method is presented for analyzing and reengineering class hierarchies. In this approach, a class hierarchy is processed along with a set of applications. A fine-grained analysis of the access and subtype relationships between objects and class members is performed. The result of this analysis is again a class hierarchy.

guaranteed to be behaviorally equivalent to the original hierarchy, but in only contains the members that are req ...


Keywords: class hierarchy reengineering, concept analysis

5 Inter-class def-use analysis with partial class representations

Amie L. Souter, Lori L. Pollock, Dixie Hisley

September 1999 **ACM SIGSOFT Software Engineering Notes , Proceed
ACM SIGPLAN-SIGSOFT workshop on Program ana
tools and engineering PASTE '99**, Volume 24 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(1.27 MB\)](#) Additional Information: [full citation](#), [abst](#)
[citations](#), [index ter](#)



Object-oriented program design promotes the reuse of code not only thro
polymorphism, but also through building server classes which can be use
client classes. Research on static analysis of object-oriented software has
addressing the new features of classes, inheritance, polymorphism, and c
This paper demonstrates how exploiting the nature of object-oriented de
enable development of scalable static analyses. We p ...

6 On the syllogistic structure of object-oriented programming

Derek Rayside, Kostas Kontogiannis

July 2001 **Proceedings of the 23rd International Conference on Softwa**

Publisher: IEEE Computer Society

Full text available:  [pdf\(138.07 KB\)](#) 

[Publisher
Site](#)

Additional Information: [full citation](#), [abst](#)
[citations](#), [index ter](#)


*Recent works by Sowa and by Rayside & Campbell demonstrate that the
connection between object-oriented programming and the logical forma
syllogism, first set down by Aristotle in the Prior Analytics. In this paper
understanding of polymorphic method invocations in terms of the syllogi
understanding to the design of a novel editor for object-oriented program
able to display a polymorphic call graph, which ...*

7 A study of dead data members in C++ applications

◆ Peter F. Sweeney, Frank Tip

May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN on Programming language design and implementation PLDI**
Issue 5

Publisher: ACM Press

Full text available:  [pdf\(1.05 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)


Object-oriented applications may contain data members that can be removed from an application without affecting program behavior. Such "dead" data members arise from unused functionality in class libraries, or due to the programmer losing track of data members as usage as the application changes over time. We present a simple and efficient algorithm for detecting dead data members in C++ applications. This algorithm has been implemented using a prototype version of the IBM VisualAge C++ compiler, ...

8 A member lookup algorithm for C++

◆ G. Ramalingam, Harini Srinivasan

May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN on Programming language design and implementation PLDI**
Issue 5

Publisher: ACM Press

Full text available:  [pdf\(1.53 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The member lookup problem in C++ is the problem of resolving a specific member access in the context of a specified class. Member lookup in C++ is complicated by features such as virtual inheritance and multiple inheritance. In this paper, we present an efficient algorithm for member lookup in C++. We also present a formalism for the multiple inheritance mechanism of C++, which we use as the basis for deriving our algorithm. This formalism may also be of use as a formal basis for deriving other C++ ...

9 Technical papers: software maintenance: Concern graphs: finding and describing structural program dependencies

◆ Martin P. Robillard, Gail C. Murphy

May 2002 **Proceedings of the 24th International Conference on Software Engineering**

Publisher: ACM Press

Full text available:  [pdf\(1.35 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

MB)citings, index ter

Many maintenance tasks address concerns, or features, that are not well represented in the source code comprising a system. Existing approaches available to help locate and manage scattered concerns use a representation based on lines of code, complicating the analysis of the concerns. In this paper, we introduce the representation that abstracts the implementation details of a concern and the relationships between different parts of the code ...

10 Fast static analysis of C++ virtual function calls

◆ David F. Bacon, Peter F. Sweeney

October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM conference on Object-oriented programming, systems, languages and applications OOPSLA '96**, Volume 31 Issue 10

Publisher: ACM Press

Full text available:  [pdf\(2.10 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citings](#), [index terms](#)


Virtual functions make code easier for programmers to reuse but also make it harder for compilers to analyze. We investigate the ability of three static analysis algorithms to improve C++ programs by resolving virtual function calls, thereby reducing code size and reducing program complexity so as to improve both human and machine understanding and analysis. In measurements of seven programs of significant size (20000 lines of code each) we found that on average the most effective algorithm ...

11 Practical experience with an application extractor for Java

◆ Frank Tip, Chris Laffra, Peter F. Sweeney, David Streeter

October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM conference on Object-oriented programming, systems, languages and applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Full text available:  [pdf\(2.31 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citings](#), [index terms](#)

Java programs are routinely transmitted over low-bandwidth network connections as compressed class file archives (i.e., zip files and jar files). Since archive size is proportional to download time, it is desirable for applications to be as small as possible. This paper is concerned with the use of program transformations such as removing unused methods and fields, inlining of method calls, and simplification of the class hierarchy.


reducing application size. Such “extract ...

12 Beyond traditional program slicing

Anthony M. Sloane, Jason Holdsworth

May 1996 **ACM SIGSOFT Software Engineering Notes , Proceedings of SIGSOFT international symposium on Software testing and '96**, Volume 21 Issue 3

Publisher: ACM Press


Full text available:  [pdf\(726.83 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

Traditional program slices are based on variables and statements. Slices are statements that potentially affect (or are affected by) the value of a particular given statement. Two assumptions are implicit in this definition: 1) that statements are concepts of the programming language in which the program is written 2) that slices consist solely of statements. Generalised slicing is an extension of traditional slicing where variables are replaced by arbitrary expressions ...

13 A schema for interprocedural modification side-effect analysis with pointers

Barbara G. Ryder, William A. Landi, Philip A. Stocks, Sean Zhang, Rita A. Kent
March 2001 **ACM Transactions on Programming Languages and Systems**, Volume 23 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(1.72 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

The first interprocedural modification side-effects analysis for C (MODC) that achieves better than worst-case precision on programs with general-purpose pointer usage. Empirical results. The analysis consists of an algorithm schema corresponding to MODC algorithms with two independent phases: one for determining pointer aliases and a subsequent one for propagating interprocedural ...

14 A simple graph-based intermediate representation

Cliff Click, Michael Paleczny

March 1995 **ACM SIGPLAN Notices , Papers from the 1995 ACM SIGPLAN conference on Intermediate representations**, Volume 30 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(1.39 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

MB)terms


We present a graph-based intermediate representation (IR) with simple memory-cost C++ implementation. The IR uses a directed graph with labeled inputs but unordered outputs. Vertices are labeled with opcodes, unlabeled. We represent the CFG and basic blocks with the same vertex. Each opcode is defined by a C++ class that encapsulates opcode-specific behavior. We use inheritance to abstract common opcode behavior ...

15 Efficient subtyping tests with PQ-encoding

◆ Yoav Zibin, Joseph Yossi Gil

October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM conference on Object oriented programming, systems, languages and applications OOPSLA '01**, Volume 36 Issue 11

Publisher: ACM Press

Full text available:  [pdf\(226.81 KB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)


Subtyping test, i.e., determining whether one type is a subtype of another operation during the execution of object-oriented programs. The challenge is to maintain a hierarchy in a small space, while simultaneously making sure that subtyping is efficient implementation. We present a new scheme for encoding multiple inheritance hierarchies, which, in the standardization of hierarchies, reduces the space previously published schemes. The scheme is ...

16 Logical foundations of object-oriented and frame-based languages

◆ Michael Kifer, Georg Lausen, James Wu

July 1995 **Journal of the ACM (JACM)**, Volume 42 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(7.52 MB\)](#) Additional Information: [full citation](#), [abstracts](#), [citations](#), [index terms](#)

We propose a novel formalism, called Frame Logic (abbr., F-logic), that can be used in declarative fashion for most of the structural aspects of object-oriented languages. These features include object identity, complex objects, inheritance, types, query methods, encapsulation, and others. In a sense, F-logic stands in a close relationship to the object-oriented paradigm as classical predicate calculus stands to relational programming. ...

Keywords: deductive databases, frame-based languages, logic programming, inheritance, object-oriented programming, proof theory, semantics, typing

17 The design space layer: supporting early design space exploration for core-


◆ Helvio P. Peixoto, Margarida F. Jacome, Ander Royo, Juan C. Lopez
January 1999 **Proceedings of the conference on Design, automation and**
Publisher: ACM Press

Full text available:  pdf(83.19 KB) Additional Information: [full citation](#), [index terms](#)

18 A debate on language and tool support for design patterns

◆ Craig Chambers, Bill Harrison, John Vlissides
January 2000 **Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on**
of programming languages

Publisher: ACM Press

Full text available:  pdf(2.04 MB) Additional Information: [full citation](#), [abstract](#), [index terms](#)


Design patterns have earned a place in the developer's arsenal of tools for software development. They have proved so useful, in fact, that some have been promoted to programming language features. In turn this has rekindled the debate over the mechanism that belong in programming languages versus those served by tools. The debate comes full circle when one contemplates the methodological tool support for patterns. The author ...

Keywords: design patterns, programming languages, software development

19 Hierarchies and relative operators in the OLAP environment

◆ Elaheh Pourabbas, Maurizio Rafanelli
March 2000 **ACM SIGMOD Record**, Volume 29 Issue 1

Publisher: ACM Press

Full text available:  pdf(585.68 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

In the last few years, numerous proposals for modelling and querying M

Databases (MDDDB) are proposed. A rigorous classification of the different hierarchies is still an open problem. In this paper we propose and discuss types of hierarchies within a single dimension of a cube. These hierarchies at different levels of aggregation a single dimension. Depending on them, we characterize some OLAP operators that refer to hierarchie ...

20 The GOLD definition language (GDL): an object oriented formal specification of multidimensional databases



Juan Trujillo, Manuel Palomar, Jaime Gómez

March 2000 **Proceedings of the 2000 ACM symposium on Applied computing**

Publisher: ACM Press

Full text available: [pdf\(421.67 KB\)](#) Additional Information: [full citation](#), [reference index terms](#)

Keywords: OLAP, conceptual modeling, data warehouses, multidimensional databases, object-orientation

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#)

The ACM Portal is published by the Association for Computing Machinery
ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#)

[Sign in](#)[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [Maps](#) [more](#)[Ac](#)
[Pr](#)

Web Results 1 - 10 of about 129,000 for **slicing class + optimizing**. (0.55 sec)

Photoshop and ImageReady Training

Slicing in Photoshop and ImageReady. Understanding Slices; **Slice** Types; **Slice** Lines ... Webucator offers very affordable at-your-own-pace online **classes** on ...

www.webucator.com/WebDesign/PsWeb.cfm - 22k - [Cached](#) - [Similar pages](#)

Optimizing RAM Memory Demand

More than 95% of the **class** file data could be removed such that the resulting ... Jamaica uses one additional thread to allow time **slicing** between threads. ...

www.aicas.com/jamaica/doc/html/x1258.html - 18k - [Cached](#) - [Similar pages](#)

198:198:671 Seminar in Object-oriented Programming Languages

F. Tip, JD Choi, J. Field, G. Ramalingam, "**Slicing Class** Hierarchies", ... "The Jalapeno Dynamic **Optimizing** Compiler for Java", 1999 ACM Java Grande ...

www.cs.rutgers.edu/~ryder/oosem99/topics.html - 22k - [Cached](#) - [Similar pages](#)

198:516

Building and **optimizing** basic blocks, recovering code from expression DAGs, ... **Class** DG for OO **slicing** References: A. Rountev, A. Milanova, BG Ryder, ...

www.cs.rutgers.edu/~ryder/516/sp06/lectures/index.html - 12k - [Cached](#) - [Similar pages](#)

IBM Research | | jdchoi | Patents

System and method for **optimizing** computer code using a compact data flow ... Method and apparatus for **slicing class** hierarchies. US Patent 6179491. ...

domino.research.ibm.com/comm/

research_people.nsf/pages/jdchoi.patents.html - 14k - [Cached](#) - [Similar pages](#)

Macromedia Fireworks Training Course

Who Should Attend: This **class** is designed for Web designers and graphics ...
Creating image maps • **Optimizing** and exporting graphics • **Slicing** images ...
www.ledet.com/outlines/html/fireworks8.shtml - 58k - [Cached](#) - [Similar pages](#)

Adobe Photoshop ImageReady Training Course

In this course you will save files in common web formats and
optimize them for quick ... Prerequisites: Because of the amount of
material we cover in **class**, ...

www.ledet.com/outlines/html/photoshopimageready.shtml - 61k -
[Cached](#) - [Similar pages](#)

[[More results from www.ledet.com](#)]

CUIN 7317, The Visual Representation of Information

Optimizing Animated GIFs and Minimizing Color Palettes Available online:
... **Slicing** larger graphics and then animating sections Examples: ...
www.coe.uh.edu/courses/cuin7317/students/class11/class11.html - 13k -
[Cached](#) - [Similar pages](#)

Image Slicing - How and Why to Slice Web Images

Simply **slicing** and saving each section with the same compression settings is
pointless; but by **optimizing**, you can reduce your images by several KB's
and ...

graphicssoft.about.com/od/webgraphics/a/whyslice.htm - 28k -
[Cached](#) - [Similar pages](#)

Training Macromedia Fireworks AZ in New York NY New York

Slice complex images into sections and use the **Optimize** panel and ... and
other job related **classes** in areas like Office Administration and Bookkeeping,
...

[www.training-classes.com/course_hierarchy/
courses/7004_Macromedia_Fireworks_A-Z.php](http://www.training-classes.com/course_hierarchy/courses/7004_Macromedia_Fireworks_A-Z.php) - 22k -
[Cached](#) - [Similar pages](#)

Try your search again on [Google Book Search](#)

Goooooooooooooogle ►

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

slicing class + optimizing

Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2006 Google

[Sign in](#)[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [Maps](#) [more](#)[Ac](#)
[Pr](#)

Web Results 1 - 10 of about **80,500** for **unused data item hierarchical merg**

[PPT] [A Unified Framework for Schedule and Storage Optimization](#)

File Format: Microsoft Powerpoint - [View as HTML](#)

Consider **merging** two stages in a program ? L C S. StreamIt: **hierarchical** stream ... At this time, **data items** are routed using a simple dimension-ordered ...

cag.csail.mit.edu/~mgordon/raw_backend.ppt - [Similar pages](#)

[PDF] [Bandwidth Skimming: A Technique for Cost-Effective Video-on-Demand](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

data required to effect the **merge** is received, and that each individual **item** is received in time to meet its viewing deadline. ...

www.cs.washington.edu/homes/zahorjan/papers/bandwidth-skimming.pdf - [Similar pages](#)

[The Alsoft Collection: a review of Alsoft's power utilities](#)

4. MenuExtend (v 1.00) This little gem makes the System 7 Apple menu **hierarchical**. When you drag down to an Apple menu **item**, if it is a folder, ...

www.savetz.com/ku/ku/duroux_alsoft_collection.html - 16k -

[Cached](#) - [Similar pages](#)

[Index](#)

hiding factory methods; **hierarchical** file system (HFS). compiler **data** sets · defining file ... length of **data items**, finding; LENGTH OF special register ...

publib.boulder.ibm.com/infocenter/

pdthelp/topic/com.ibm.entcobol4.doc/pgindex.htm - 377k -

[Cached](#) - [Similar pages](#)

[PDF] [Code and Toggle Coverage](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

(mind the differences related to collecting **hierarchical data** and **data** for units ... It helps to verify the quality of the stimulus and locate **unused** ...

www.aldec.com/training/riviera/2006.02/pdf/C07_Code_Coverage.pdf - [Similar pages](#)

[PDF] Efficient Simulation for Hierarchical and Partitioned Circuits

File Format: PDF/Adobe Acrobat

of the **unused** circuits. Each time a new set of operands. is supplied to one unit, ... contains several **data items**, including a pointer to the ...

doi.ieeecomputersociety.org/10.1109/ICVD.1999.745154 - [Similar pages](#)

Xcell Journal Online -- Pin Assignment article Issue 55

PlanAhead software provides a complete environment to make this **hierarchical** methodology interactive and easy to use by presenting design **data** through the ...

www.xilinx.com/publications/xcellonline/xcell_55/xc_planahead55.htm - 34k - [Cached](#) - [Similar pages](#)

[DOC] Survivability

File Format: Microsoft Word 2000 - [View as HTML](#)

Are there **unused** or unexpected components? This provides an opportunity for ... **Data items** of interest are the Web pages manipulated by GeoWorlds (both ...

www.objs.com/DASADA/CDSA2001.doc - [Similar pages](#)

Driver Guide - free downloads - freeware and shareware Finance ...

Optimize your expenditure based on current income **data**. ... s unique recursive tree view is a new paradigm for organizing **hierarchical data** it is hands-down ...

drn.digitalriver.com/category.php%5Baction%5Dbrowse&i=30&id=115&f=%7C%7C%7C%7C&s=product.... - 62k - [Cached](#) - [Similar pages](#)

Internet › HTML Tools- Free Downloads at Simtel.net

94% of computers have corrupt, **unused**, and possibly harmful files. Run a FREE registry scan to clean, repair, and **optimize** your system with the leading and ...

www.simtel.net/category.php%5Baction%5Dbrowse&i=20&id=152&f=%7C%7C%7C%7C&s=product.date_r... - 70k - [Cached](#) - [Similar pages](#)

Try your search again on [Google Book Search](#)

Goooooooooooooogle ►

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Free! Speed up the web. [Download the Google Web Accelerator.](#)

unused data item hierarchical merge **Search**

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2006 Google



optimize unused data objects merge

1990

- 2

Scholar Results 1 - 10 of about 976 for optimize unused data objects merge

Iterative type analysis and extended message splitting: Optimizing dynamically-typed object-oriented ... - group of 14 »

[All articles](#) [Recent articles](#)

CJ Chambers, DJ Ungar - Higher-Order and Symbolic Computation, 1991 - Springer

... the compiler is frequently able to **optimize** away the ... Standard **data** flow techniques,

on the other hand ... important for dynami- cally-typed **object**-oriented languages ...

[Cited by 98](#) - [Web Search](#)

[PS] The Design and Implementation of the Self Compiler, an Optimizing Compiler for Object-Oriented ... - group of 7 »

C Chambers - 1992 - research.sun.com

... 13.1.4 Interactions between Debugging and **Optimization** ... terms of implementations of

existing abstract **data** types, and ... to invoke operations on **objects** of unknown ...

[Cited by 126](#) - [View as HTML](#) - [Web Search](#) - [Library Search](#)

Indexing and querying XML **data** for regular path expressions - group of 41 »

Q Li, B Moon - ... of the 27th International Conference on Very Large **Data** ..., 2001 - nike.psu.edu

... necessary until all the reserved spaces (ie, **unused** or- der ... traverse the hi-erarchy

of XML **objects** in either ... and bottom-up approaches for XML **data** of certain ...

[Cited by 373](#) - [View as HTML](#) - [Web Search](#) - [BL Direct](#)

Honey, I shrunk the XQuery!: an XML algebra optimization approach - group of 3 »

X Zhang, B Pielech, EA Rundesnteiner - ... fourth international workshop on Web information and **data** ..., 2002 - portal.acm.org

... and relational **data** in the loading manager ... XAT, introduced in Section 2, for **optimization** and execution ... see Figure 9). XAT Cleaner eliminates **unused** columns in ...

[Cited by 26](#) - [Web Search](#)

[INFRASTRUCTURE, QUERY OPTIMIZATION, **DATA** WAREHOUSING AND **DATA** MINING FOR SCIENTIFIC SIMULATION - group of 2 »](#)

Y Huang - 2002 - nd.edu

... 30 4.1 **Objects**, Reactions and Processes 78 6.4 Populate The **Data** Warehouse 81 6.5 Query **Optimization**

[Cited by 2](#) - [View as HTML](#) - [Web Search](#) - [Library Search](#)

[Whole-Program Optimization of Object-Oriented Languages - group of 5 »](#)

JA Dean - 1996 - marvin.kset.org

... away this flexibility when it is **unused** by a ... class testing allows the compiler to

optimize message sends ... **data** to determine where it is profitable to compile ...

[Cited by 28](#) - [View as HTML](#) - [Web Search](#) - [Library Search](#)

[Marmot: an optimizing compiler for Java - group of 17 »](#)

R Fitzgerald, TB Knoblock, E Ruf, B Steensgaard, D ... - Software-Practice and Experience, 2000 - doi.wiley.com

... prior to conversion to remove **unused** methods from ... This **optimization** is deferred until

after register allocation ... the Marmot libraries and their **data** layouts are ...

[Cited by 116](#) - [Web Search](#) - [BL Direct](#)

[**Data** and memory optimization techniques for embedded systems](#)

PR Panda, F Catthoor, ND Dutt, K Danckaert, E ... - ACM Transactions on Design Automation of Electronic Systems ..., 2001 - portal.acm.org

... the code using a loop- **merging** transformation, gives ... as merely geometrical **objects**,

without execution ... is particularly important to **optimize data** transfers and ...

[Cited by 99](#) - [Web Search](#) - [BL Direct](#)

[Whole-program optimization of object-oriented languages - group of 10 »](#)

C Chambers, J Dean, D Grove - University of Washington Seattle, Technical Report 96-06, 1996 - research.ibm.com

... a closure is stored in a **data** structure that ... **Unused** potential will be optimized

automatically and simply ... When attempting to **optimize** a dynamic dispatch, we ...

[Cited by 56](#) - [View as HTML](#) - [Web Search](#)

[A Portable Approach to Dynamic Optimization in Run-time Specialization - group of 4 »](#)

H Masuhara, A Yonezawa - New Generation Computing, 2001 - graco.cu-tokyo.ac.jp

... in Proceedings of Second Symposium on Programs as **Data Objects** (PADO-II ... Some

optimization algorithms are ... live local variables, and the types of **unused** ones do ...

[Cited by 5](#) - [View as HTML](#) - [Web Search](#) - [BL Direct](#)

Go o o o o o o o o o o g l e ►

Result Page: 1 2 3 4 5 6 7 8 9 10 [Next](#)

optimize unused data objects merge

Search

[Google Home](#) - [About Google](#) - [About Google Scholar](#)

©2006 Google



Microsoft Corporation

[Subscribe \(Full Service\)](#) [Register \(Limited Service\)](#)
Search: ☐ The ACM Digital Library ☐ The ACM

THE GUIDE TO COMPUTING LITERATURE

[Feedback](#) [Report a problem](#)
Slicing class hierarchies in C++**Full text** [Pdf \(2.08 MB\)](#)**Source** [Conference on Object Oriented Programming Systems Language archive](#)
Proceedings of the 11th ACM SIGPLAN conference on Object-programming, systems, languages, and applications [table of contents](#)
 San Jose, California, United States

Pages: 179 - 197

Year of Publication: 1996

ISSN:0362-1340

[Also published in ...](#)
Authors [Frank Tip](#) IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA
[Jong-Deok Choi](#) IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA
[John Field](#) IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA
[G. Ramalingam](#) IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, USA
Sponsor [SIGPLAN](#): ACM Special Interest Group on Programming Language**Publisher** ACM Press New York, NY, USA
Additional Information: [abstract](#) [references](#) [citations](#) [index terms](#) [collaborative peer review](#)
Tools and Actions: [Find similar Articles](#) [Review this Article](#)
[Save this Article to a Binder](#) Display Formats: [BibTeX](#)
DOI Bookmark: Use this link to bookmark this Article: <http://doi.acm.org/10.1145/236337.236355>
[What is a DOI?](#)
↑ **ABSTRACT**

This paper describes an algorithm for *slicing* class hierarchies in C++ program hierarchy (a collection of C++ classes and inheritance relations among them)

uses the hierarchy, the algorithm eliminates from the hierarchy those data members, functions, classes, and inheritance relations that are unnecessary for ensuring that the hierarchy is maintained. Class slicing is especially useful when the program P is generated from program P' by a *statement slicing* algorithm. Such an algorithm eliminates statements that are irrelevant to a set of slicing *criteria*---program points of particular interest. This is a considerable improvement over previous work on statement slicing, and it will not be the concern of this paper. However, the combination of statement slicing and class slicing for C++ has two important applications: First, class slicing can enhance statement slicing's utility in program understanding applications, by eliminating both executable *and* declarative program elements that are irrelevant to the slicing criteria. Second, the combination of the two slicing algorithms can decrease the space requirements of programs that do not use all the components of a class library. Such a situation is particularly common in programs that use class libraries.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text. We have opted to expose the complete List rather than only correct and linked references.

- 1 ACCREDITED STANDARDS COMMITTEE X3, I. P. S. Working paper for the development of an international standard for information systems---programming language C++. 1995.
- 2 AGESEN, O. Constraint-based type inference and parametric polymorphism. First International Static Analysis Symposium (SAS'94) (September 1994), 78-91. LNCS vol. 864.
- 3 Ole Agesen, Concrete type inference: delivering object-oriented applications, Stanford University, Stanford, CA, 1996
- 4 Ole Agesen , David Ungar, Sifting out the gold: delivering compact applications in an exploratory object-oriented programming environment, ACM SIGPLAN Notices 370, Oct. 1994
- 5 Hiralal Agrawal, On slicing programs with jump statements, Proceedings of the 1994 conference on Programming language design and implementation, p.302-311, 1994, Orlando, Florida, United States
- 6 Hiralal Agrawal , Richard A. DeMillo , Eugene H. Spafford, Dynamic slicing of programs with unconstrained pointers, Proceedings of the symposium on Testing, analysis, and

73, October 08-10, 1991, Victoria, British Columbia, Canada

7 David F. Bacon , Peter F. Sweeney, Fast static analysis of C++ virtual functions of the 11th ACM SIGPLAN conference on Object-oriented programming, systems applications, p.324-341, October 06-10, 1996, San Jose, California, United States

8 BACON, D. F., WEGMAN, M., AND ZADECK, F. K. Rapid type inference. RC 1234, IBM Thomas J. Watson Research Center, 1995.

9 Thomas Ball , Susan Horwitz, Slicing Programs with Arbitrary Control-flow. First International Workshop on Automated and Algorithmic Debugging, p.20-30, 1993

10 Brad Calder , Dirk Grunwald, Reducing indirect function call overhead in C. Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.397-408, January 16-19, 1994, Portland, Oregon, United States

11 CARINI, P. R., HIND, M., AND SRINIVASAN, H. Flowsensitive type inference. Rep. RC 20267, IBM T.J. Watson Research Center, 1995.

12 Jong-Deok Choi , Jeanne Ferrante, Static slicing in the presence of goto statements. Transactions on Programming Languages and Systems (TOPLAS), v.16 n.4, 1994

13 DEAN, j., AND CHAMBERS, C. Optimization of objectoriented programs using a control flow hierarchy analysis. Tech. Rep. 94-12-01, Department of Computer Science, University of Washington at Seattle, December 1994.

14 Margaret A. Ellis , Bjarne Stroustrup, The annotated C++ reference manual. Longman Publishing Co., Inc., Boston, MA, 1990

15 ERNST, M. Practical fine-grained static slicing of optimized code. Tech. Rep. Microsoft Research, Redmond, WA, 1994.

16 Mary F. Fernández, Simple and effective link-time optimization of Modula-2. Proceedings of the ACM SIGPLAN 1995 conference on Programming language implementation, p.103-115, June 18-21, 1995, La Jolla, California, United States

17 John Field , G. Ramalingam , Frank Tip, Parametric program slicing, Proceedings of the ACM SIGPLAN-SIGACT symposium on Principles of programming languages, p.23-25, 1995, San Francisco, California, United States

- 18 Susan Horwitz , Thomas Reps , David Binkley, Interprocedural slicing us graphs, ACM Transactions on Programming Languages and Systems (TOPLAS), v.12 n.1, p.1-41, Jan. 1990
- 19 KRISHNASWAM~, A. Program slicing: An application of object-oriented graphs. Technical report TR94-108, Dept. of Computer Science, Clemson Un
- 20 Loren Larsen , Mary Jean Harrold, Slicing object-oriented software, Proc international conference on Software engineering, p.495-505, March 25-29, 1
- 21 LIVADAS, P. E., AND CROLL, S. Program slicing. Report serc-tr-61-f, Department, University of Florida, 1992.
- 22 Jens Palsberg , Michael I. Schwartzbach, Object-oriented type inference, Notices, v.26 n.11, p.146-161, Nov. 1991
- 23 PANDE, H. D., AND RYOER, B. G. Static type determination and aliasi LCSR-TR-250-A, Rutgers University, October 1995.
- 24 John Plevyak , Andrew A. Chien, Precise concrete type inference for obje ACM SIGPLAN Notices, v.29 n.10, p.324-340, Oct. 1994
- 25 Thomas Reps , Susan Horwitz , Mooly Sagiv , Genevieve Rosay, Speedin Proceedings of the 2nd ACM SIGSOFT symposium on Foundations of softwa 20, December 06-09, 1994, New Orleans, Louisiana, United States
- 26 Jonathan G. Rossie, Jr. , Daniel P. Friedman, An algebraic semantics of s SIGPLAN Notices, v.30 n.10, p.187-199, Oct. 17, 1995
- 27 SAKKINEN, M. A critique of the inheritance principles of C++. Comput (1992), 69--110.
- 28 Amitabh Srivastava, Unreachable procedures in object-oriented program Programming Languages and Systems (LOPLAS), v.1 n.4, p.355-364, Dec. 1
- 29 TIP, F. A survey of program slicing techniques. Journal of Programming 121-189.
- 30 Mark David Weiser, Program slices: formal, psychological, and practical automatic program abstraction method, 1979

↑ CITINGS 17

Bi-Xin Li , Xiao-Cong Fan , Jun Pang , Jian-Jun Zhao, A model for slicing JA hierarchically, Journal of Computer Science and Technology, v.19 n.6, p.848-

G. Ramalingam , Harini Srinivasan, A member lookup algorithm for C++, AC v.32 n.5, p.18-30, May 1997

Amie L. Souter , Lori L. Pollock , Dixie Hisley, Inter-class def-use analysis w representations, ACM SIGSOFT Software Engineering Notes, v.24 n.5, p.47-

Derek Rayside , Kostas Kontogiannis, On the syllogistic structure of object-o Proceedings of the 23rd international conference on Software engineering, p.1 2001, Toronto, Ontario, Canada

Frank Tip , Chris Laffra , Peter F. Sweeney , David Streeter, Practical experie extractor for Java, ACM SIGPLAN Notices, v.34 n.10, p.292-305, Oct. 1999

Peter F. Sweeney , Frank Tip, A study of dead data members in C++ applicati Notices, v.33 n.5, p.324-332, May 1998

David F. Bacon , Peter F. Sweeney, Fast static analysis of C++ virtual functio SIGPLAN Notices, v.31 n.10, p.324-341, Oct. 1996

Frank Tip , Peter F. Sweeney, Class hierarchy specialization, ACM SIGPLAN p.271-285, Oct. 1997

Martin P. Robillard , Gail C. Murphy, Concern graphs: finding and describing structural program dependencies, Proceedings of the 24th international confer engineering, May 19-25, 2002, Orlando, Florida

Gregor Snelting , Frank Tip, Reengineering class hierarchies using concept ar SIGSOFT Software Engineering Notes, v.23 n.6, p.99-110, Nov. 1998

Katsuhisa Maruyama , Ken-Ichi Shima, An Automatic Class Generation Mec Method Integration, IEEE Transactions on Software Engineering, v.26 n.5, p.

Derek Rayside , Kostas Kontogiannis, Extracting Java library subsets for depl systems, Science of Computer Programming, v.45 n.2-3, p.245-270, Novemb

Gregor Snelting , Frank Tip, Understanding class hierarchies using concept analysis, Transactions on Programming Languages and Systems (TOPLAS), v.22 n.3, June 2001

Frank Tip , Peter F. Sweeney , Chris Laffra , Aldo Eisma , David Streeter, Practical
techniques for Java, ACM Transactions on Programming Languages and Systems
v.22 n.6, p.625-666, November 2002

Ramkrishna Chatterjee , Barbara G. Ryder , William A. Landi, Complexity of
Java in the Presence of Exceptions, IEEE Transactions on Software Engineering
v.29 n.5, p.411-424, June 2001

Barbara G. Ryder , William A. Landi , Philip A. Stocks , Sean Zhang , Rita A.
interprocedural modification side-effect analysis with pointer aliasing, ACM
Transactions on Programming Languages and Systems (TOPLAS), v.23 n.2, p.105-186, March 2002

Baowen Xu , Ju Qian , Xiaofang Zhang , Zhongqiang Wu , Lin Chen, A brief
slicing, ACM SIGSOFT Software Engineering Notes, v.30 n.2, March 2005

↑ INDEX TERMS

Primary Classification:

D. Software

↳ D.1 PROGRAMMING TECHNIQUES

Additional Classification:

D. Software

↳ D.3 PROGRAMMING LANGUAGES

↳ D.3.2 Language Classifications

↳ Nouns: C++

↳ D.3.3 Language Constructs and Features

↳ Subjects: Data types and structures

F. Theory of Computation

↳ F.3 LOGICS AND MEANINGS OF PROGRAMS

↳ F.3.3 Studies of Program Constructs

↳ Subjects: Type structure

General Terms:Algorithms, Languages, Performance, Theory**↑ Collaborative Colleagues:**

<u>Jong-Deok Choi:</u>	<u>Bowen Alpern</u> <u>Koenraad De Bosschere</u> <u>Michael Burke</u> <u>Michael G. Burke</u> <u>Paul Carini</u> <u>Paul R. Carini</u> <u>Soumen Chakrabarti</u> <u>Mark Charney</u> <u>Mark Christiaens</u> <u>Anthony Cocchi</u> <u>R. Cytron</u> <u>Ron Cytron</u> <u>J. Ferrante</u>	<u>Jeanne Ferrante</u> <u>John Field</u> <u>Stephen Fink</u> <u>David Grove</u> <u>Manish Gupta</u> <u>Michael Hind</u> <u>Clinton Jeffery</u> <u>William G. Landi</u> <u>Keunwoo Lee</u> <u>Raimondas Lencevicius</u> <u>Derek Lieber</u> <u>Alexey Loginov</u> <u>Thomas J. Marlowe</u>	<u>Sam Midkiff</u> <u>Samuel P. Midkiff</u> <u>B. P. Miller</u> <u>Barton P. Mill</u> <u>Sang L. Min</u> <u>Sang Lyul Mi</u> <u>Robert H. B. Netzer</u> <u>Ton Ngo</u> <u>Robert O'Callahan</u> <u>G. Ramalinga</u> <u>Michiel Ronss</u> <u>Barbara G. Ry</u> <u>Vivek Sarkar</u>
<u>John Field:</u>	<u>Jan A. Bergstra</u> <u>Jong-Deok Choi</u> <u>T. B. Dinesh</u> <u>Deepak Goyal</u> <u>Jan Heering</u> <u>Roman Manevich</u> <u>G. Ramalingam</u> <u>Mooly Sagiv</u> <u>Bruce Segee</u> <u>Gregor Snelting</u>	<u>Tim Teitelbaum</u> <u>Frank Tip</u> <u>Carlos A. Varela</u> <u>Alex Warshavsky</u>	
<u>G. Ramalingam:</u>	<u>Jong-Deok Choi</u> <u>John Field</u> <u>G. Goos</u> <u>Deepak Goyal</u> <u>J. Hartmanis</u> <u>Michelle Y. Kim</u>	<u>Thomas Reps</u> <u>Thomas W. Reps</u> <u>Mooly Sagiv</u> <u>Junehwa Song</u> <u>Harini Srinivasan</u> <u>Frank Tip</u>	

	<u>J. van Leeuwen</u>	<u>Alex Warshavsky</u>	
	<u>Roman Manevich</u>	<u>Eran Yahav</u>	
	<u>Raymond Miller</u>	<u>Byoung-Kee Yi</u>	
	<u>C. Pandu Rangan</u>		
<u>Frank Tip:</u>	<u>Dirk Bäumer</u>	<u>Robert Fuhrer</u>	<u>Fenil Shah</u>
	<u>Ittai Balaban</u>	<u>Markus Keller</u>	<u>Gregor Sneltir</u>
	<u>Thomas Ball</u>	<u>Adam Kiezun</u>	<u>Maximilian</u>
	<u>A. Michael Berman</u>	<u>Adam Kiezun</u>	<u>Stoerzer</u>
	<u>Ophelia Chesley</u>	<u>Paul Klint</u>	<u>David Streeter</u>
	<u>Jong-Deok Choi</u>	<u>Chris Laffra</u>	<u>Peter F. Sweet</u>
	<u>Arie Deursen</u>	<u>Jens Palsberg</u>	
	<u>T. B. Dinesh</u>	<u>G. Ramalingam</u>	
	<u>Aldo Eisma</u>	<u>Xiaoxia Ren</u>	
	<u>John Field</u>	<u>Barbara G. Ryder</u>	

↑ **Peer to Peer - Readers of this Article have also read:**




- Data structures for quadtree approximation and compression **Communications of the ACM** 28, 9
Hanan Samet
- A hierarchical single-key-lock access control using the Chinese remainder of the 1992 ACM/SIGAPP Symposium on Applied computing
Kim S. Lee , Huizhu Lu , D. D. Fisher
- The GemStone object database management system **Communications of the ACM** 34, 12
Paul Butterworth , Allen Otis , Jacob Stein
- An intelligent component database for behavioral synthesis **Proceedings of the ACM/IEEE conference on Design automation**
Gwo-Dong Chen , Daniel D. Gajski
- Putting innovation to work: adoption strategies for multimedia communication **Communications of the ACM** 34, 12
Ellen Francik , Susan Ehrlich Rudman , Donna Cooper , Stephen Levine

↑ **This Article has also been published in:**

- **ACM SIGPLAN Notices**
Volume 31 , Issue 10 Oct. 1996

The ACM Portal is published by the Association for Computing Machinery
ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Med
Player](#)